

LIN 127

Text Processing and Corpus Linguistics

Week 8

Python for Text

Timeline for the Rest of the Quarter!

- Today:
 - Assignment 5 Notes
 - External Libraries in Python
 - Assignment 6 Preview (due 3/8)
- Next Week: regular expressions, more on NLP libraries in Python, research chat

Assignment 5 Notes

- Readability - “Pythonic”?
 - `' '.join([c for c in s if c.isalpha()])`
 - Could be written out as a for loop, etc
 - Personal preference!
 - (it is mine though)

Assignment 5 Notes

- word_counts
 - Note that the Counter object can take a list of objects to count as part of its constructor
 - To discuss - what is a constructor?

Assignment 5 Notes

- syllable_count, final_syllable
 - Follow the given definition!
Space-separated phoneme ending in a digit.
 - You can “make it work” other ways,
but might run into hard-to-foresee edge cases

Assignment 5 Notes

- words_rhyme
 - How do we check if *any* pair of possible pronunciations rhyme?
 - Nested for loops!
 - Put the return statement in the deepest point

Assignment 5 Notes

- flesch_reading_ease
 - We want to calculate two quantities:
 - average_line_length
 - average_syllables_per_word
 - For average syllables, what do we have to say about words for which we don't have pronunciation?
 - Nothing!

Assignment 5 Notes

- flesch_reading_ease
 - Further word on this:
is this actually “readability”?
 - It is validated to some degree,
and correlates with human judgments of readability
 - Nevertheless, it is an oversimplification -
often we say, a particular “operationalization”
 - Always look at these critically!!!
Are we missing important things in your context?

Assignment 5 Notes

- Calculating adjacent lines that rhyme
 - Requires some more complex “spatial awareness”
 - Let’s work through it together

Assignment 5 Notes

- Calculating delta dictionaries
 - Use subtract method on Counter!
 - Have to look it up - note that it is “in-place” and does not return anything

External Libraries for Text Processing

- A6 looking at spaCy and NLTK
- All kinds of useful functionality!

NAME	DESCRIPTION
Tokenization	Segmenting text into words, punctuations marks etc.
Part-of-speech (POS) Tagging	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “rats” is “rat”.
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named “real-world” objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a knowledge base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.

External Libraries for Text Processing

- In this assignment we'll be using their tokenizer, stemmer, lemmatizer, and part-of-speech tagger
- Always important to refer to documentation when using external libraries!

Stemming and Lemmatization

- Forms of text normalization, reduces sparsity

Stemming vs Lemmatization

change
changing
changes
changed
changer



The diagram shows five arrows pointing from the words 'change', 'changing', 'changes', 'changed', and 'changer' to the stem 'chang'.

chang

change
changing
changes
changed
changer



The diagram shows five arrows pointing from the words 'change', 'changing', 'changes', 'changed', and 'changer' to the lemma 'change'.

change

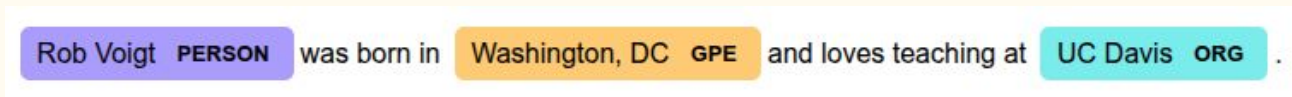
Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

<https://www.turing.com/kb/stemming-vs-lemmatization-in-python>

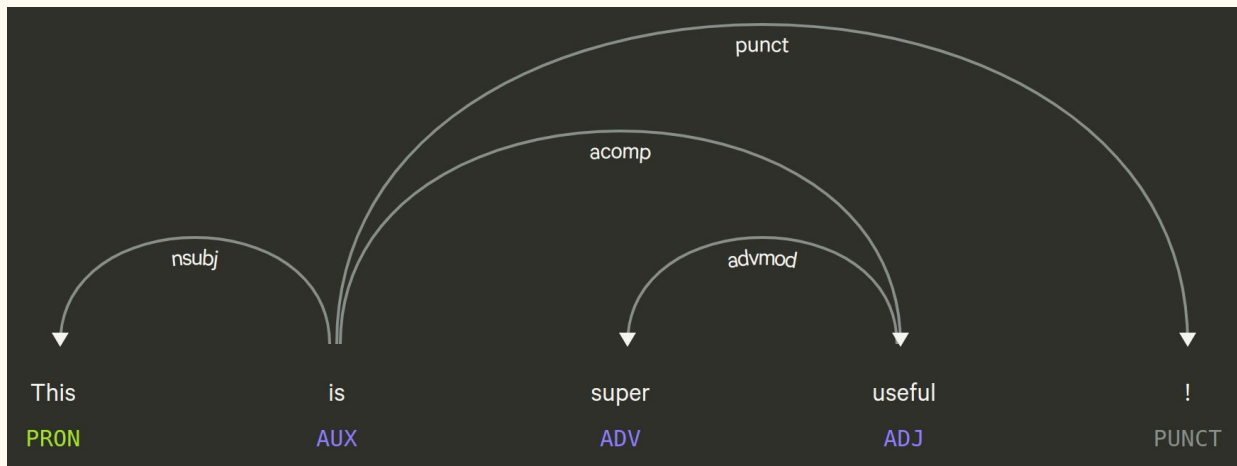
<https://studymachinelearning.com/stemming-and-lemmatization/>

External Libraries for Text Processing

- More complex functionality often useful, like entity tagging:



- and dependency parsing:



Assignment 6 Preview

- Let's check it out!